

REMARKS

Applicant respectfully requests reconsideration and allowance of the present application. Claims 1-25, 27-31, 34-35, and 37-43 are pending in this application. Claims 1, 11, 20, 28, 35 and 40 are independent claims.

Applicant's amendments and remarks after Final are appropriate under 37 C.F.R. §1.116 because they address the Office's remarks in the Final Action, and thus could not have been presented earlier. In addition, the amendments and remarks should be entered to place the case in better form for appeal.

Claim Rejection Under 35 U.S.C. § 103

Claims 1-25, 27-31, 34-35, and 37-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,336,139 to Feridun et al. (hereinafter "Feridun") in view of U.S. Patent No. 6,751,753 to Nguyen et al. (hereinafter "Nguyen"). The rejection is respectfully traversed.

Applicant has determined that the Nguyen patent may be sworn behind by submitting an affidavit or declaration in accordance with 37 CFR 1.131. However, at this time, Applicant elects to argue over the current rejection under 35 U.S.C. 103(a). Applicant retains the right to swear behind the Nguyen patent in the future and intends to do so if the 35 U.S.C. 103(a) is maintained.

I. Introduction

The present Application describes implementations that provide a flexible correlation system and/or method that allows a user to correlate multiple events and/or data. One benefit of the implementations is that a non-programmer or application developer can make use of predefined correlation functions that monitor user selected events and/or data parameters.

1 In one implementation, a state machine is used to simultaneously monitor
2 different events and data. The state machine may be a class object. The class
3 object may have a number of properties that are modifiable by a non-programmer.
4 For example, the class object may include a string that represents the name of a
5 process being monitored; a string that holds an integer that identifies the number
6 of crashes of the process that will trigger an event if those crashes occur within a
7 particular time period; and a string that holds another integer that defines the
8 particular time period. Therefore, a non-programmer can define such a class
9 object my simply filling-in the values for the process, the number of crashes and
10 the particular time period.

11 The non-programmer can also create an updating consumer used to update
12 the state of the state machine described in the above-paragraph. The updating
13 consumer is also a class object. The updating consumer class object is separate
14 from the state machine. In one example, the updating consumer updates an
15 instance of the state machine when a crash occurs. Therefore, the updating
16 consumer ensures that the state machine maintains an up-to-date number of
17 crashes that has been observed by the state machine.

18 The foregoing discussion is not limiting of the claimed invention. Instead,
19 the discussion was provided to enhance the Office's understanding of at least one
20 implementation described in the present Application.
21
22
23
24
25

II. Rejection Discussion

Claim 1 recites:

A computer-implemented method comprising:
receiving a plurality of events;
applying the plurality of events to a correlation function, wherein the correlation function is implemented as a state machine and is configured to correlate the plurality of events;
identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine;
applying the update consumer to the state machine in response to the identified event; and
generating a specific event if the correlation function is satisfied by the plurality of events.

The Feridum patent describes the use of correlation rules implemented as state machines to recognize patterns of one or more events indicative of given conditions. According to Feridum, the observed events are examined to determine if one or more defined event patterns has occurred. If an event pattern is recognized, an event correlator may be used to take a defined action. (See Feridum, column 2, lines 42-62.)

The Feridum patent does not teach or suggest at least “identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine; [and] applying the update consumer to the state machine in response to the identified event”, as recited in claim 1. The Office realizes that the Feridum patent does not teach at least these limitations of claim 1 and attempts to cure this by combining Feridum with Nguyen.

The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-40, and column 6, lines 26-35 of Nguyen teaches the indicated limitations of claim 1. The Applicant disagrees for the following reasons.

1 Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen
2 describe a status object that includes current information related to a state of a
3 monitored system component. The status object receives information pertaining to
4 the monitored system component and transitions its state if a transition condition
5 associated with the status object is satisfied. If this occurs, some action is taken by
6 the status object. (See generally, column 2, lines 1-14.) There is nothing in this
7 section of the Nguyen patent that teaches or suggests the limitations from claim 1
8 that include: “identifying an event to which an update consumer has subscribed,
9 wherein the update consumer is associated with the state machine; [and] applying
10 the update consumer to the state machine in response to the identified event.”

11 The Nguyen patent further describes the use of a state machine class and a
12 status object class. The status object class is updated if a transition state
13 associated with an evaluation function linked to the state machine class is
14 satisfied. Therefore, the status object class can be used to identify a state of the
15 state machine class. (See generally, column 2, lines 24-40.) This teaching of the
16 Nguyen patent does not approach the limitations of claim 1 that recite:
17 “identifying an event to which an update consumer has subscribed, wherein the
18 update consumer is associated with the state machine; [and] applying the update
19 consumer to the state machine in response to the identified event.” More
20 specifically, there is nothing in the Nguyen patent that indicates that the status
21 object class is *subscribed* to an event, as recited in claim 1. At most, the status
22 object class in Nguyen monitors a state of the state machine class. This is not the
23 same as being subscribed to an event. Moreover, how can it be fairly said that the
24 status object is *applied* to the state machine in response to the identified event?
25 The Applicant submits that it cannot be fairly said as such. The status object class

1 in Nguyen simply monitors the state machine class; monitoring is not the same as
2 something being applied.

3 Column 6, lines 26-40 of Nguyen fails to teach or suggest the above-
4 discussed limitations of claim 1. This portion of Nguyen describes the use of an
5 action 130 designed to implement an action when a state of the state machine class
6 changes. This teaching has nothing to do with “identifying an event to which an
7 update consumer has subscribed, wherein the update consumer is associated with
8 the state machine; [and] applying the update consumer to the state machine in
9 response to the identified event”, as recited in claim 1.

10 In accordance with the above, Applicant submits that the Nguyen patent
11 does not cure the deficiencies of the Feridum patent. Accordingly, at least claim 1
12 is not rendered obvious by the Feridum and Nguyen combination.

13 **Claim 11** recites:

14 A computer-implemented method comprising:
15 receiving a plurality of events;
16 receiving a plurality of data elements;
17 identifying a plurality of correlation functions configured to
18 correlate the plurality of events and the plurality of data elements, wherein
19 each correlation function is implemented with an associated state machine,
20 and wherein each state machine has an associated update consumer that
21 updates the state of the associated state machine;
22 applying the plurality of events and the plurality of data elements to
23 the plurality of correlation functions; and
24 generating a specific event if at least one of the plurality of
25 correlation functions is satisfied.

21 The Feridum patent describes the use of correlation rules implemented as
22 state machines to recognize patterns of one or more events indicative of given
23 conditions. According to Feridum, the observed events are examined to determine
24 if one or more defined event patterns has occurred. If an event pattern is
25

1 recognized, an event correlator may be used to take a defined action. (See
2 Feridum, column 2, lines 42-62.)

3 The Feridum patent does not teach or suggest at least “each state machine
4 has an associated update consumer that updates the state of the associated state
5 machine”, as recited in claim 11. The Office realizes that the Feridum patent does
6 not teach at least the indicated limitation of claim 11 and attempts to cure this by
7 combining Feridum with Nguyen.

8 The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-
9 40, and column 6, lines 26-35 of Nguyen teaches the indicated limitation of claim
10 11. The Applicant disagrees for the following reasons.

11 Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen
12 describe a status object that includes current information related to a state of a
13 monitored system component. The status object receives information pertaining to
14 the monitored system component and transitions its state if a transition condition
15 associated with the status object is satisfied. If this occurs, some action is taken by
16 the status object. (See generally, column 2, lines 1-14.) There is nothing in this
17 section of the Nguyen patent that teaches or suggests the limitation from claim 11
18 that includes: “each state machine has an associated update consumer that updates
19 the state of the associated state machine.”

20 The Nguyen patent further describes the use of a state machine class and a
21 status object class. The status object class is updated if a transition state
22 associated with an evaluation function linked to the state machine class is
23 satisfied. Therefore, the status object class can be used to identify a state of the
24 state machine class. (See generally, column 2, lines 24-40.) This teaching of the
25 Nguyen patent does not approach the limitation of claim 11 that recites: “each

1 state machine has an associated update consumer that updates the state of the
2 associated state machine.” More specifically, there is nothing in the Nguyen
3 patent that indicates that the status object class *updates the state* of the state
4 machine class, as recited in claim 11. At most, the status object in Nguyen
5 monitors the state machine class to determine when the state machine class
6 updates. However, that act of monitoring in Nguyen does not teach or suggest that
7 “each state machine has an associated update consumer that updates the state of
8 the associated state machine”, as recited in claim 11.

9 Column 6, lines 26-40 of Nguyen fails to teach or suggest the above-
10 discussed limitations of claim 11. This portion of Nguyen describes the use of an
11 action 130 designed to implement an action when a state of the state machine class
12 changes. This teaching has nothing to do with “each state machine has an
13 associated update consumer that updates the state of the associated state machine”,
14 as recited in claim 11.

15 In accordance with the above, Applicant submits that the Nguyen patent
16 does not cure the deficiencies of the Feridum patent. Accordingly, at least claim
17 11 is not rendered obvious by the Feridum and Nguyen combination.

18 **Claim 20** recites:

19 A computer-implemented method comprising:
20 identifying a schema for creating state machines, the state machines
21 to correlate at least two events;
22 creating an instance of a particular state machine;
23 defining transitions for the particular state machine by subscribing to
24 at least one event; and
25 applying an update consumer to the particular state machine to
update the state of the particular state machine, wherein the update
consumer is a class object.

1 The Feridum patent describes the use of correlation rules implemented as
2 state machines to recognize patterns of one or more events indicative of given
3 conditions. According to Feridum, the observed events are examined to determine
4 if one or more defined event patterns has occurred. If an event pattern is
5 recognized, an event correlator may be used to take a defined action. (See
6 Feridum, column 2, lines 42-62.)

7 The Feridum patent does not teach or suggest at least “applying an update
8 consumer to the particular state machine to update the state of the particular state
9 machine, wherein the update consumer is a class object”, as recited in claim 20.
10 The Office realizes that the Feridum patent does not teach at least this limitation
11 of claim 20 and attempts to cure this by combining Feridum with Nguyen.

12 The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-
13 40, and column 6, lines 26-35 of Nguyen teaches the indicated limitation of claim
14 20. The Applicant disagrees for the following reasons.

15 Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen
16 describes a status object that includes current information related to a state of a
17 monitored system component. The status object receives information pertaining to
18 the monitored system component and transitions its state if a transition condition
19 associated with the status object is satisfied. If this occurs, some action is taken by
20 the status object. (See generally, column 2, lines 1-14.) There is nothing in this
21 section of the Nguyen patent that teaches or suggests the limitation from claim 20
22 that includes: “applying an update consumer to the particular state machine to
23 update the state of the particular state machine, wherein the update consumer is a
24 class object”, as recited in claim 20.
25

1 The Nguyen patent further describes the use of a state machine class and a
2 status object class. The status object class is updated if a transition state
3 associated with an evaluation function linked to the state machine class is
4 satisfied. Therefore, the status object class can be used to identify a state of the
5 state machine class. (See generally, column 2, lines 24-40.) This teaching of the
6 Nguyen patent does not approach the limitation of claim 20 that recites: “applying
7 an update consumer to the particular state machine to update the state of the
8 particular state machine, wherein the update consumer is a class object.” More
9 specifically, there is nothing in the Nguyen patent that indicates that the status
10 object class is *applied* to the state machine class to *update* the state of the state
11 machine class, as recited in claim 20. At most, the status object class in Nguyen
12 *monitors* the state machine class to identify when the state machine class
13 undergoes a state change. Monitoring the state machine in Nguyen is not the same
14 as actively participating in causing the state machine to change states, as recited in
15 claim 20. And, in addition, the state machine class in Nguyen changes its state on
16 its own. There is no compelling reason to think that Nguyen would use the status
17 object class to update a state of the state machine class.

18 Column 6, lines 26-40 of Nguyen fails to teach or suggest the above-
19 discussed limitations of claim 20. This portion of Nguyen describes the use of an
20 action 130 designed to implement an action when a state of the state machine class
21 changes. This teaching has nothing to do with “applying an update consumer to
22 the particular state machine to update the state of the particular state machine,
23 wherein the update consumer is a class object”, as recited in claim 20.
24
25

1 In accordance with the above, Applicant submits that the Nguyen patent
2 does not cure the deficiencies of the Feridum patent. Accordingly, at least claim
3 20 is not rendered obvious by the Feridum and Nguyen combination.

4 **Claim 28** recites:

5 An apparatus comprising:
6 a plurality of event consumers; and
7 an event correlator coupled to the plurality of event consumers, the
8 event correlator to receive events from at least one event source and to
9 receive data elements from at least one data source, the event correlator
10 further to receive at least one correlation function configured to correlate
11 events and data elements and to apply the received events and the received
12 data elements to the correlation function, wherein the correlation function is
13 implemented by a state machine having an associated update consumer that
14 updates the state of the state machine, and wherein the event correlator
15 generates a specific event if the received events and the received data
16 satisfy the correlation function.

17 Claim 28 includes the limitation “wherein the correlation function is
18 implemented by a state machine having an associated update consumer that
19 updates the state of the state machine.” For the same reasons presented in
20 connection with the response to the rejection of claim 20, Applicant submits at
21 least claim 28 is not rendered obvious by the Feridum and Nguyen combination.

22 **Claim 35** recites:

23 One or more computer-readable media having stored thereon a
24 computer program that, when executed by one or more processors, causes
25 the one or more processors to:
receive a plurality of events;
identify a plurality of correlation functions configured to correlate
the plurality of events, wherein each of the plurality of correlation functions
is implemented as a state machine having an associated update consumer;
apply the plurality of events to the plurality of correlation functions
to determine whether any of the plurality of correlation functions are
satisfied by the plurality of events, wherein the plurality of events are
applied by causing update consumers associated with each event to update
the state of the associated state machine; and

1 generate a specific event if one of the plurality of correlation
2 functions is satisfied by the plurality of events.

3 Claim 35 includes the limitation “wherein the plurality of events are
4 applied by causing update consumers associated with each event to update the
5 state of the associated state machine.” For the same reasons presented in
6 connection with the response to the rejection of claims 20 and 28, Applicant
7 submits at least claim 35 is not rendered obvious by the Feridum and Nguyen
8 combination.

9 **Claim 40** recites:

10 A computer-implemented method comprising:
11 receiving events from event providers;
12 creating a first state machine;
13 creating a second state machine;
14 associating a first event type with the first state machine, wherein the
15 first state machine has an associated first update consumer to update the
16 state of the first state machine;
17 associating a second event type with the second state machine,
18 wherein the second state machine has an associated second update
19 consumer to update the state of the second state machine;
20 in response to receiving an event having a first event type, applying
21 the first update consumer to the first state machine;
22 in response to receiving an event having a second event type,
23 applying the second update consumer to the second state machine; and
24 if the events are correlated:
25 generating an additional event; and
sending the additional event to an event consumer.

19 The Feridum patent describes the use of correlation rules implemented as
20 state machines to recognize patterns of one or more events indicative of given
21 conditions. According to Feridum, the observed events are examined to determine
22 if one or more defined event patterns has occurred. If an event pattern is
23 recognized, an event correlator may be used to take a defined action. (See
24 Feridum, column 2, lines 42-62.)
25

1 The Feridum patent does not teach or suggest at least “wherein the first
2 state machine has an associated first update consumer to update the state of the
3 first state machine;... wherein the second state machine has an associated second
4 update consumer to update the state of the second state machine;... in response to
5 receiving an event having a first event type, applying the first update consumer to
6 the first state machine; [and] in response to receiving an event having a second
7 event type, applying the second update consumer to the second state machine”, as
8 recited in claim 40. The Office realizes that the Feridum patent does not teach at
9 least these limitations of claim 40 and attempts to cure this by combining Feridum
10 with Nguyen.

11 The Office asserts that column 1, lines 58-62, column 2, lines 1-14 and 35-
12 40, and column 6, lines 26-35 of Nguyen teaches the indicated limitations of claim
13 40. The Applicant disagrees for the following reasons.

14 Column 1, lines 58-62, and column 2, lines 1-14 and 35-40 of Nguyen
15 describes a status object that includes current information related to a state of a
16 monitored system component. The status object receives information pertaining to
17 the monitored system component and transitions its state if a transition condition
18 associated with the status object is satisfied. If this occurs, some action is taken by
19 the status object. (See generally, column 2, lines 1-14.) There is nothing in this
20 section of the Nguyen patent that teaches or suggests the limitations from claim 40
21 that include: “wherein the first state machine has an associated first update
22 consumer to update the state of the first state machine;... wherein the second state
23 machine has an associated second update consumer to update the state of the
24 second state machine;... in response to receiving an event having a first event type,
25 applying the first update consumer to the first state machine; [and] in response to

1 receiving an event having a second event type, applying the second update
2 consumer to the second state machine.”

3 The Nguyen patent further describes the use of a state machine class and a
4 status object class. The status object class is updated if a transition state
5 associated with an evaluation function linked to the state machine class is
6 satisfied. Therefore, the status object class can be used to identify a state of the
7 state machine class. (See generally, column 2, lines 24-40.) This teaching of the
8 Nguyen patent does not approach the limitations of claim 40 that recite: “wherein
9 the first state machine has an associated first update consumer to update the state
10 of the first state machine;... wherein the second state machine has an associated
11 second update consumer to update the state of the second state machine;... in
12 response to receiving an event having a first event type, applying the first update
13 consumer to the first state machine; [and] in response to receiving an event having
14 a second event type, applying the second update consumer to the second state
15 machine.”

16 More specifically, there is nothing in the Nguyen patent that indicates that
17 the status object class *updates* the state of the state machine class, as recited in
18 claim 40. At most, the status object class in Nguyen *monitors* the state machine
19 class to identify when the state machine class undergoes a state change.
20 Monitoring the state machine in Nguyen is not the same as actively participating
21 in causing the state machine to change states, as recited in claim 40. And, in
22 addition, the state machine class changes its state on its own. Therefore, there is
23 no compelling reason to think that Nguyen would use the status object class to
24 update a state of the state machine class.
25

1 Column 6, lines 26-40 fails to teach or suggest the above-discussed
2 limitations of claim 40. This portion of Nguyen describes the use of an action 130
3 designed to implement an action when a state of the state machine class changes.
4 This teaching has nothing to do with “wherein the first state machine has an
5 associated first update consumer to update the state of the first state machine;...
6 wherein the second state machine has an associated second update consumer to
7 update the state of the second state machine;... in response to receiving an event
8 having a first event type, applying the first update consumer to the first state
9 machine; [and] in response to receiving an event having a second event type,
10 applying the second update consumer to the second state machine”, as recited in
11 claim 40.

12 In accordance with the above, Applicant submits that the Nguyen patent
13 does not cure the deficiencies of the Feridum patent. Accordingly, at least claim
14 40 is not rendered obvious by the Feridum and Nguyen combination.

15 For the reasons presented above, Applicant respectfully submits claims 1,
16 11, 20, 28, 35 and 40 are at least allowable over the Feridum and Nguyen
17 combination. The remaining dependent claims are allowable by virtue of their
18 dependency on one of the discussed independent claims.
19
20
21
22
23
24
25

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

8

9

10

10

11

12